# Inferring nucleosome positions with their histone mark annotation from ChIP data
Supplementary Material

Alessandro Mammana [1]*, Martin Vingron [2] and Ho-Ryun Chung [1]
Max Planck Institute for Molecular Genetics, Ihnestrasse 63-73 D-14195 Berlin, Germany
[1]Otto-Warburg-Laboratories, Epigenomics
[2]Computational Molecular Biology
Contact: mammana@molgen.mpg.de

June 20, 2013

## Contents

# 1 Inferring the average fragment length

## 1.1 Description of the EM algorithm

The goal of the EM algorithm is to infer the average length of the fragments of a single-end sequencing experiment from a (peak) cross-correlation function in a given interval (see the main document).

Let $CC : [x_{min}, x_{max}] \to \mathbb{N}$ denote the (peak) cross-correlation function truncated to a certain interval (typically from 0 to 300). The inference procedure makes the following assumptions:

*to whom correspondence should be addressed

1. $M$ is a random variable that takes on integer values within the range $[x_{min}, x_{max}]$ and $CC(x)$ is the function that tallies the occurrences of the value $x$.

2. The random variable $M$ is a mixture of three random variables $G_1$, $G_2$ and $U$. That is, there is a random variable $K$ that can take on values 1 2 or 3 with probabilities respectively $\pi_1$, $\pi_2$ and $\pi_3$ and such that:

$$M|\{K = k\} \sim \begin{cases} G_1, & k = 1 \\ G_2, & k = 2 \\ U, & k = 3 \end{cases}.$$

   The coefficients $\pi_i$ will be referred to as the mixing coefficients. $G_1$ and $G_2$ represent respectively the phantom peak and the fragment peak (Landt *et al.*, 2012), while $U$ represents the background noise.

3. The random variables $G_i, i = 1, 2$ are distributed according to a discretized and truncated gaussian random variable with parameters $\mu_i$ and $\sigma_i$, that is:

$$Prob\{G_i = g\} = \frac{exp(-\frac{(g-\mu_i)^2}{\sigma_i^2})}{\sum_{x=x_{min}}^{x_{max}} exp(-\frac{(x-\mu_i)^2}{\sigma_i^2})}.$$

4. The random variable $U$ is uniformly distributed in the interval $[x_{min}, x_{max}]$.

From these assumptions the likelihood of $CC$ as a function of the parameters $\pi_1, \pi_2, \mu_1, \mu_2, \sigma_1, \sigma_2$ can be computed and a local maximum can be attained using the expectation maximization algorithm.

Since $G_1$ models the phantom peak commonly observed in cross-correlation analyses, the initial value for $\mu_1$ is set to the average read length, while the initial value for $\mu_2$ defaults to 147 and can be modified by the user. The initial values for $\sigma_1^2$, $\sigma_2^2$, $\pi_1$ and $\pi_2$ are respectively $36, 1000, 0.1$ and 0.1.

The phantom peak is not always present. In case the EM algorithm infers an unreasonable value for it (i.e. 20 bp apart from the average read length), the whole inference is repeated using only the components $G_2$ and $U$.

## 1.2 Analysis of the K562 dataset

We used NucHunter to compute the cross-correlation and peak cross-correlation function for the histone modification dataset for the cell line K562 (Bernstein *et al.*, 2010), followed by the inference of the average fragment length using the previously described EM algorithm. The plots in Figure 1 show for each library the two cross-correlation functions, as well as the inferred phantom and fragment peak, represented respectively by a red and green dashed lines.

The plots suggest that:

- the peak cross-correlation function is more suitable than the cross-correlation function for inferring the average fragment length because it exhibits a sharper peak
- the EM algorithm, by explicit modelling of the phantom peak, is able to identify the fragment peak correctly, even in cases where the former is higher than the latter.

## 1.3 A quality score for $\sigma$

The peak detection algorithm used by NucHunter depends mainly on the parameter $\sigma$ mentioned in the main document (see Section 2.2).

To a certain extent $\sigma$ can be chosen a priori considering how the shape of the Mexican hat wavelet depends on $\sigma$ (see Figure 2). The impulse response of the filter used for peak detection can be interpreted as a position-specific score assigned to the read counts in proximity of a candidate position (in the Figure, position 0). Ideally, the score assigned to a nucleosome should not be influenced by the read counts due to adjacent nucleosomes, which would argue for the choice of a very small $\sigma$. However, a small $\sigma$ causes an increase in false positives and less reliable nucleosome calls. A high signal-to-noise ratio should allow for smaller values of $\sigma$.

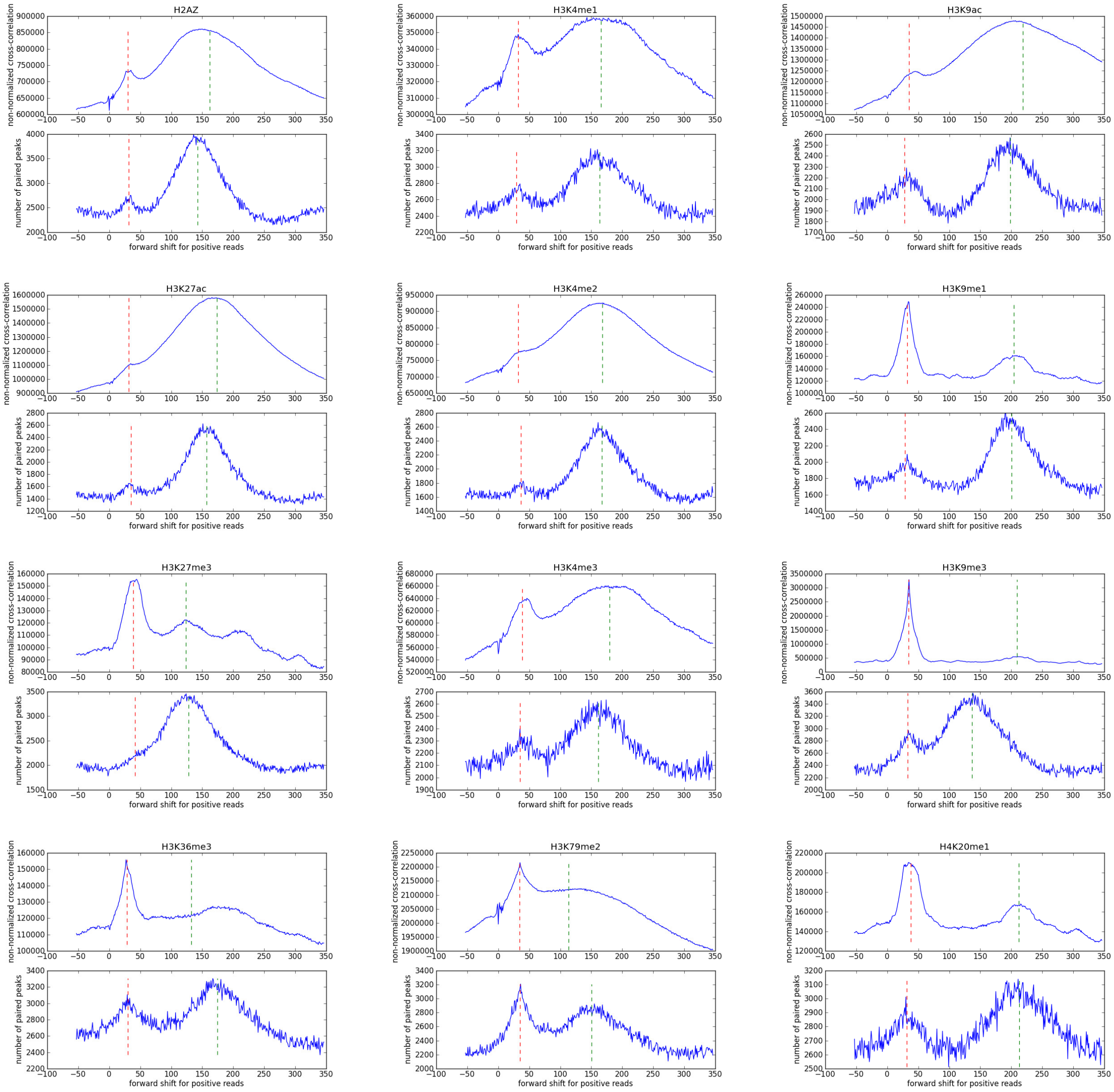


Figure 1: Inference of the average fragment length. For each histone modification ChIP-seq experiment the plot on top shows the cross-correlation function and the plots below shows the peak cross-correlation function. The green and red dashed lines represent respectively the phantom peak and the fragment peak position as inferred by the EM algorithm.

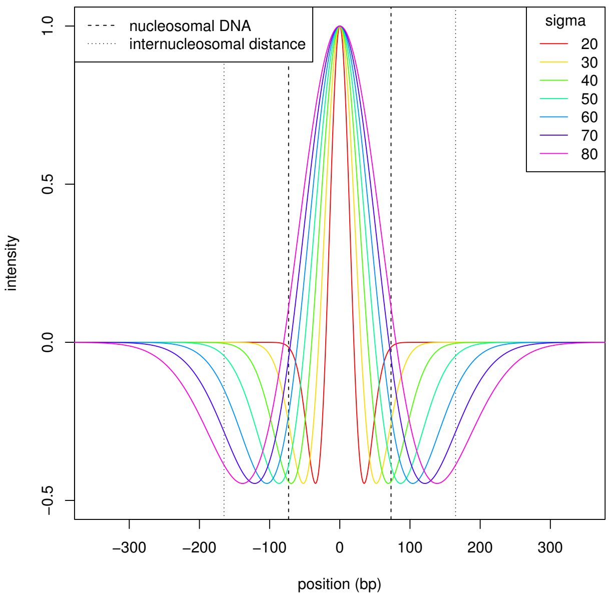


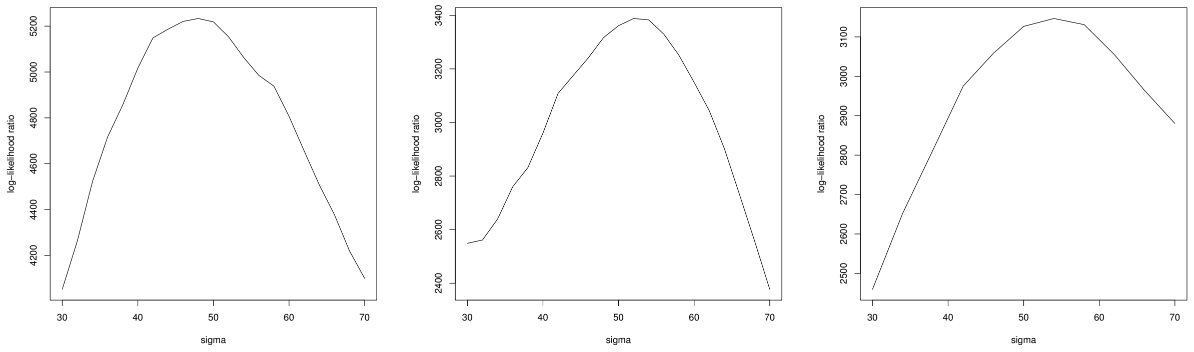
Figure 2: Dependence of the Mexican hat wavelet on $\sigma$. The wavelet can be interpreted as a position-specific score assigned to the read counts in proximity of a candidate position (in this case, position 0). The portion of DNA protected by the nucleosome is delimited by the dashed lines, the distance to the next nucleosome in case of an array of adjacent nucleosomes in yeast is shown with a dotted line. If $\sigma$ is too large, adjacent nucleosomes cannot be resolved, if it is too small, peak detection becomes too sensitive to noise.

On the other hand $\sigma$ can be chosen in a data-driven manner based on the peak cross-correlation (pcc) function. As mentioned in the main document, a strong peak in the pcc plot is also an evidence that the peaks obtained in the peak detection step are reliable. To measure the strength of the peak as a function of $\sigma$, we use the following procedure:

1. We perform peak calling on the strand-specific signals $N$ and $P$ using the algorithm outlined in Section 2.2 in the main document for different values of $\sigma$ (typically, from 30 to 70),
2. for each $\sigma$ we infer the average fragment length $F_\sigma$ from the pcc plot (which typically does not change very much),
3. we discard the lowest-scoring peaks so that for each $\sigma$ there is an equal total number of peaks from the two strands,
4. we re-compute the pcc function for the given peak set in the interval $[F_\sigma - 73, F_\sigma + 73]$ (so as to minimize the influence from adjacent nucleosomes),
5. we fit the mixture model presented in Section 1.1 constraining the mean of the peak model to $F_\sigma$ and without the phantom peak, for simplicity,
6. as a score, we consider the log-likelihood of the resulting model minus the log-likelihood of a uniform model (log-likelihood ratio).

The whole procedure is automated and parallelized and constitutes part of NucHunter. The plots in Figure 3 show how the score changes with $\sigma$ on different datasets. In all the shown examples the score curve has a maximum at a reasonable value for $\sigma$, which makes the choice easy. When this is not the case, $\sigma$ should be chosen a priori.

The default value $\sigma = 50$ is, in general, a reasonable choice, as Figure 3 suggests. Moreover Figure 4 suggests that NucHunter is sufficiently robust to sub-optimal settings of the parameters $\sigma$ and $F$.



(a) Yeast dataset presented in the main document. Maximum at $\sigma = 48$.

(b) Simulated dataset presented in Section 3. Maximum at $\sigma = 52$.

(c) Human dataset presented in Section 4. Maximum at $\sigma = 54$.

Figure 3: Quality score as a function of the parameter $\sigma$ used for peak detection. The quality score is based on the strength of the fragment peak in the peak cross-correlation plot. In the human dataset, ChIP-seq data from several histone modifications contribute to the score: the score curve has been obtained by summing up the score curves from each dataset. All the maxima occur at values close to the default value 50.

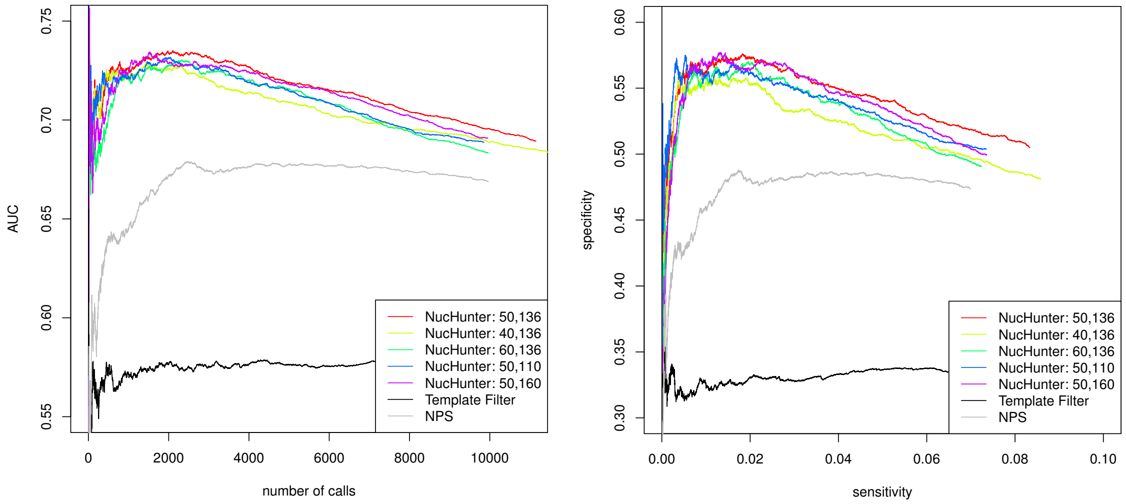


Figure 4: Robustness of NucHunter to sub-optimal settings. The performance evaluation was done in the yeast dataset and is similar to the one reported in the main document. The end points of the curves show the performance measures of the algorithms using the default score thresholds. The curves labelled by "NucHunter: $x$,$y$" show NucHunter's performance when the parameters $\sigma$ (default value: 50) and $F$ (average fragment length, estimated value on this dataset: 136) are set respectively to $x$ and $y$.

# 2 Performance measures

Given a list of high-confidence, base pair-resolution peaks $M = \{m_1, m_2, \dots m_r\}$ and a list of predicted peaks $P = \{p_1, p_2, \dots p_s\}$ we define three performance measures in order to evaluate how accurate the predicted peaks are:

1. the specificity,
2. the sensitivity,
3. the area under the (normalized) error curve (AUC).

Let $dist(i, j)$ denote the genomic distance between the predicted peak $p_i$ and the benchmark peak $m_j$. Given a cutoff distance $D$ ($D = 20$ bp in our analyses), the specificity is the quantity $\frac{|\{i:\exists j:dist(i,j)\leq D\}|}{r}$ and, similarly, the sensitivity is $\frac{|\{j:\exists i:dist(i,j)\leq D\}|}{s}$. The first performance measure does not penalize situations where many predicted peaks are close to the same benchmark peak and the second one does not penalize situations where for many closely-spaced benchmark peaks there is only one associated prediction.

In order to assess the performance of a peak caller at a higher resolution, we use a measure that depends on the distribution of the $d(i, j)$ values smaller than $W = 73$ bp (the “errors”). We define the (normalized) error curve $ce$ as the cumulative distribution function of the errors smaller than the threshold $W$:

$$ce(d) = \frac{|\{(i,j) : dist(i,j) \leq d\}|}{|\{(i,j) : dist(i,j) \leq W\}|}.$$

The cumulative error curve should look almost like a $0 - 1$ step for very precise predictions and like a straight line from the origin to the point $(W, 1)$ for random predictions (see Figure 5). Therefore, we define the area under the (normalized) error curve (AUC) as:

$$AUC = \frac{1}{W+1}\sum_{d=0}^{W} ce(d).$$

Contrary to the sensitivity and specificity, the AUC has the property that the peaks in $P$ and the peaks in $M$ play a symmetric role, i.e. swapping the predictions with the benchmark peaks the result does not change. Moreover, because it depends only on pairs of peaks closer than $W$ base pairs, the AUC is suitable for the comparison of nucleosome predictions derived from different histone marks, where a large number of peaks derived from one dataset might not have a corresponding peak derived from the other. For these reasons the AUC has been employed to compare nucleosome predictions when a nucleosome map is not available and when different histone marks are compared.

# 3 Simulated ChIP-seq experiment

As an additional test, we artificially generated a ChIP-seq sample. The simulation was done as follows.

1. We considered a chromosome of the length of chromosome IV in yeast (1531933 bp) and reads of 36 base pairs.
2. We generated reads due to noise. The number of noise reads at each genomic position was sampled from a poisson distribution with average 2.
3. We generated reads due to nucleosomes.
   - Nucleosomes were assigned to genomic positions. The positions were chosen sampling the inter-nucleosomal distance $D$ from the random variable $G + 147$, where $G$ is a geometric random variable such that $D$ averages to 165.
   - Given a fragment length $F$ (the value 140 was chosen for the simulations), and for each nucleosome position $p$, the reads on the positive and negative strands where generated sampling their positions from a gaussian random variable with average respectively $p - F/2$ and $p + F/2$

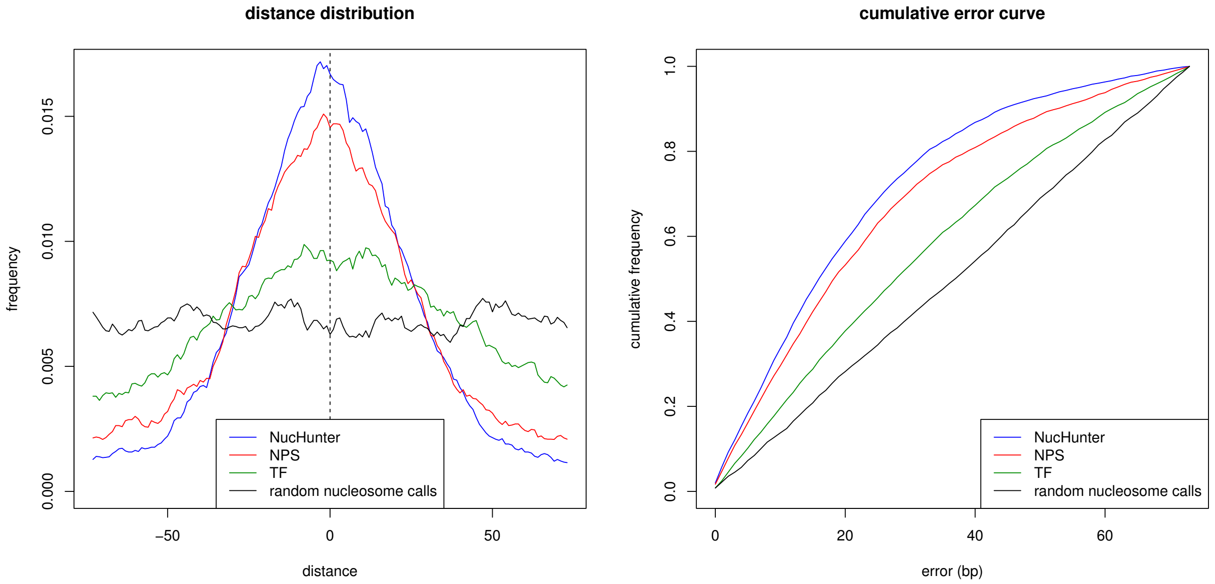


Figure 5: Distance distribution (left) and cumulative error curve (right) relative to the benchmark yeast dataset and the predictions presented in the main document. The top 5000 predictions have been chosen from each tool. The distance distribution histogram has been smoothed with a running mean in a window of 10 bps.

and with uniformly varying sigmas (from 10 to 50). The number of sampled reads was sampled from a poisson distribution with lambdas such that the expected number of reads at the peak position uniformly varies from 1 to 3.

In Figure 6 we show the performance of the three different algorithms on the the simulated dataset with respect to the performance measures outlined in Section 2.

# 4 Performance assessment on the human K562 dataset

The publicly available epigenomic data for the human cell line K562 (Bernstein *et al.*, 2010) includes ChIP-seq experiments for different histone marks as well as replicate ChIP-seq experiments. We used this dataset to test how reproducible the nucleosome calls are between replicates and to cross-validate pairs of histone marks using different nucleosome detection algorithms. For this assessment, however, it should be noted that reproducibility does not necessarily imply the reliability of the nucleosome calls, and without a high-confidence nucleosome map it is hard to draw conclusions on the performance of the algorithms, especially when the AUC values are close to 0.5.

Figures 7 and 8 show how the AUC between predictions from two ChIP-seq samples depends on the total number of nucleosome calls. In Figure 7 replicate ChIP-seq experiments have been compared, whereas in Figure 8 pairs of different histone modifications have been used. Overall the statistics suggest that, even though not for every dataset and not for every score threshold, the nucleosome predictions from NucHunter are in general more reproducible than those from other tools.

# 5 Runtime and memory usage

We tested the runtime and memory usage of the different algorithms on different datasets. We used the epigenomic data from the human cell line IMR90 made publicly available by the NIH Epigenomics Roadmap project (Bernstein *et al.*, 2010) and we performed two sets of tests. In the first set (see Table 1) we split the mapped reads from a single experiment (for histone mark H3K4me3) into different files according to the chromosome they have been mapped to (each file contains reads mapped to a single chromosome). This operation was necessary in order to test the efficiency of Template Filter, which

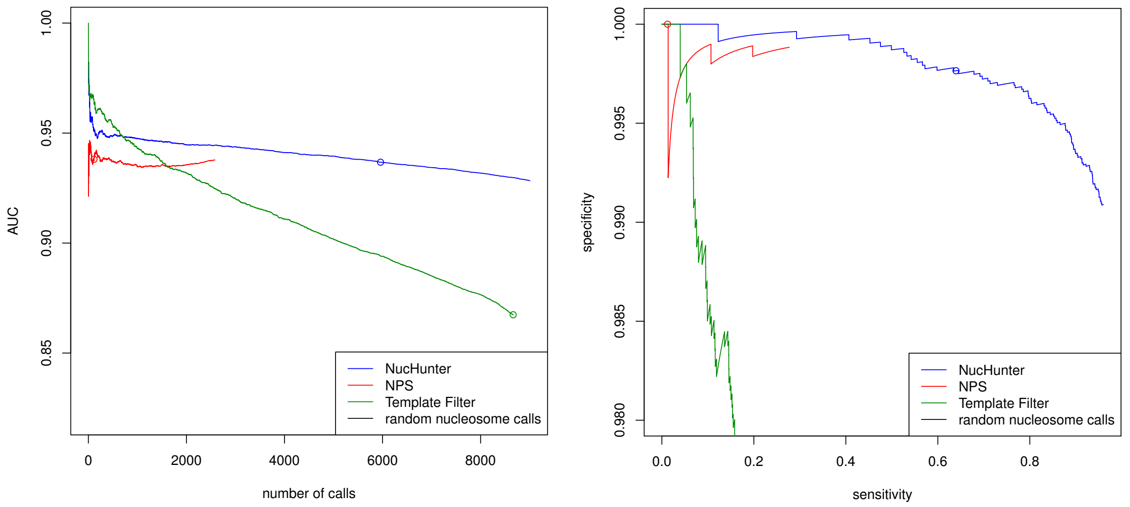


Figure 6: Performance of the three different algorithms on the simulated dataset. The circles indicate the performance of the algorithms using the default thresholds for selecting the number of peaks.

otherwise would not run. In the second set of tests Template Filter was excluded and the whole files relative to four different histone marks were used as inputs (see Table 2). All the tests were carried out on a quad-core computer at a clock speed of 3.10GHz and with 8GB of RAM. Overall the results show that NucHunter is faster than the other two algorithms and uses less memory on large genomes.

# 6 Nucleosome clustering

## 6.1 Normalization procedure

For each predicted nucleosome NucHunter returns a vector where each component represents a read count for a particular histone modification in a given window of the genome. Additionally, when a control experiment is present, NucHunter also returns the noise level, which is the read count relative to the given window and to a smoothed version of the control signal. Let $C_{ij}$ denote the read count matrix, where $i = \{1, \ldots\ n\}$ ranges over the nucleosome predictions and $j = \{1, .., m\}$ ranges over the histone marks, let $N_i$ denote the noise level for each histone modification and let $\mu$ and $\sigma$ denote the functions that compute respectively the sample mean and the sample standard deviation of a vector. The normalization procedure consists in the following steps:

1. A matrix of adjusted read count/noise level ratios $M^{(0)}_{ij} = \frac{CC_{ij}\alpha_j}{N_i}$ is computed. The histone modification-dependent coefficient $\alpha_j$ rescales the ratios so that they are concentrated around 1.

2. The matrix columns are rescaled so that they have zero mean and variance equals to one: $M^{(1)}_{ij} = \frac{M^{(0)}_{ij} - \mu(M^{(0)}_j)}{\sigma(M^{(0)}_j)}$. This steps corrects for different statistical properties of the read count signal in the different experiments.

3. The matrix rows are rescaled so that they have zero mean and variance equals to one: $M^{(2)}_{ij} = \frac{M^{(1)}_{ij} - \mu(M^{(1)}_i)}{\sigma(M^{(1)}_i)}$. This steps correct for different read abundances at different nucleosome locations.

The matrix $M^{(2)}_{ij}$ is finally used as input for the k-means clustering algorithm.

## 6.2 Clustering stability analysis

The k-means clustering algorithm is initialization-dependent. That is, given different initial values for the centroid positions, the final centroid positions might differ, especially when the parameter $k$ is not

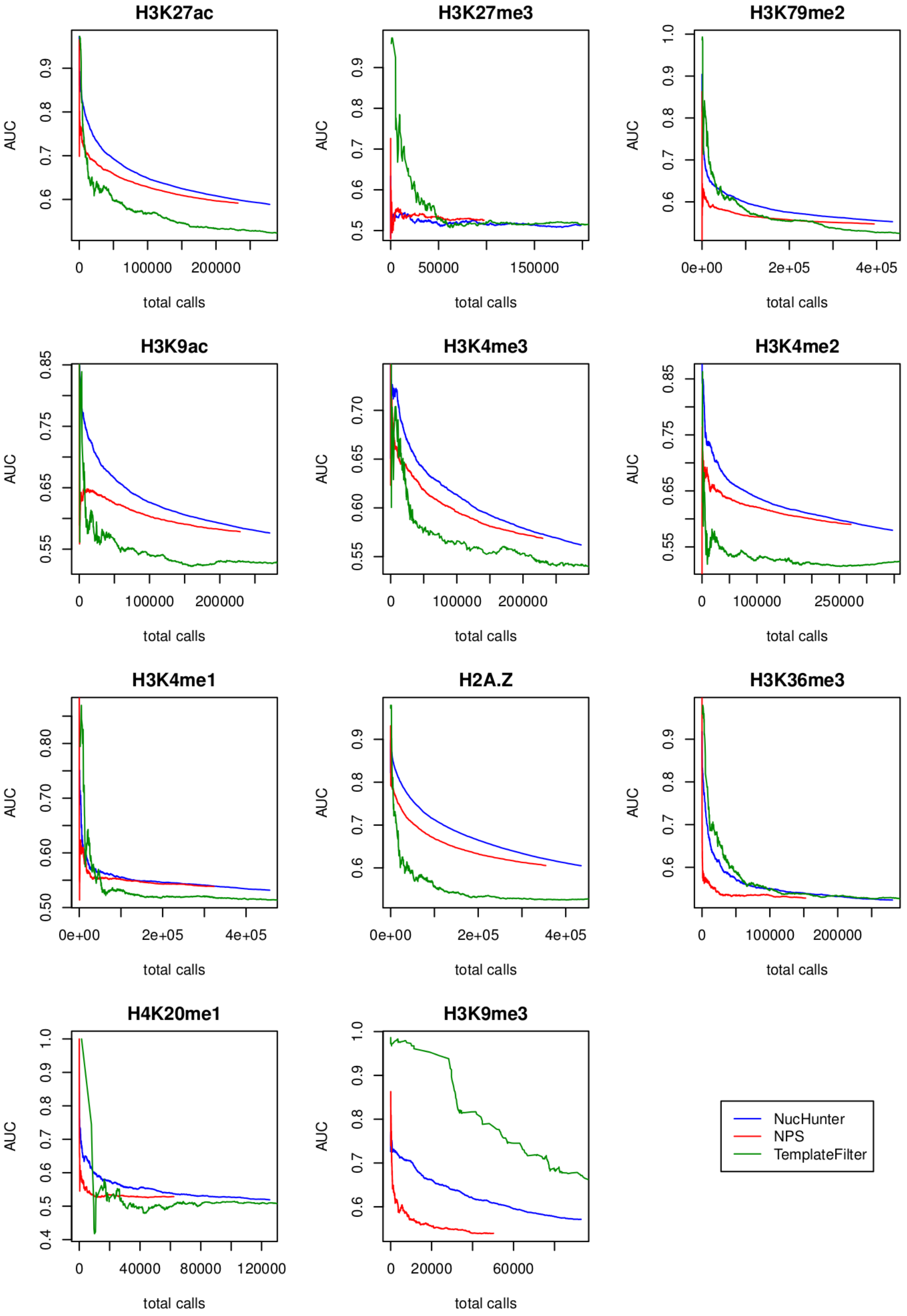


Figure 7: Performance of the three different algorithms on replicate ChIP-seq experiments. The end points of the curves show the total number of calls and the AUC using the default score thresholds.

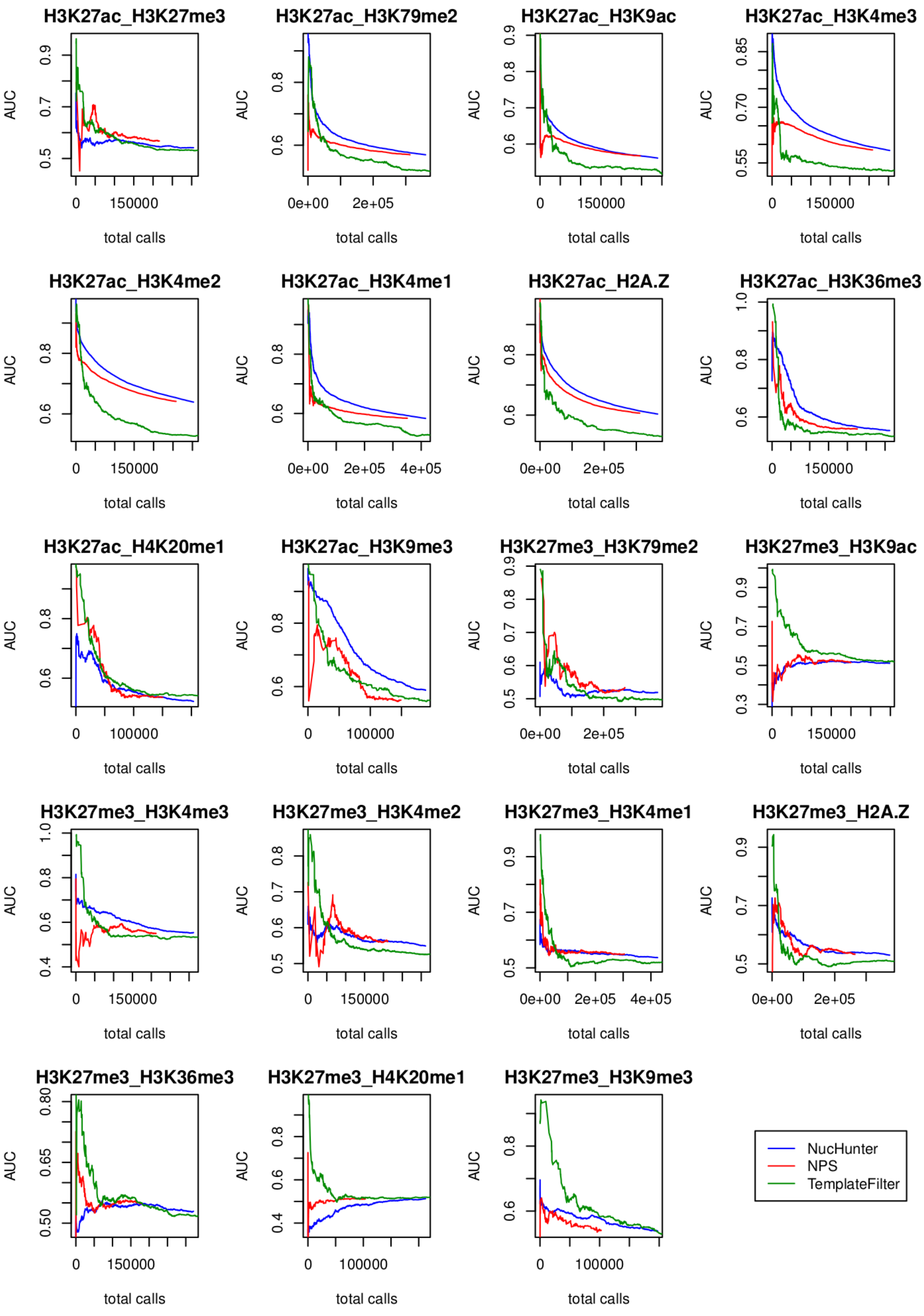
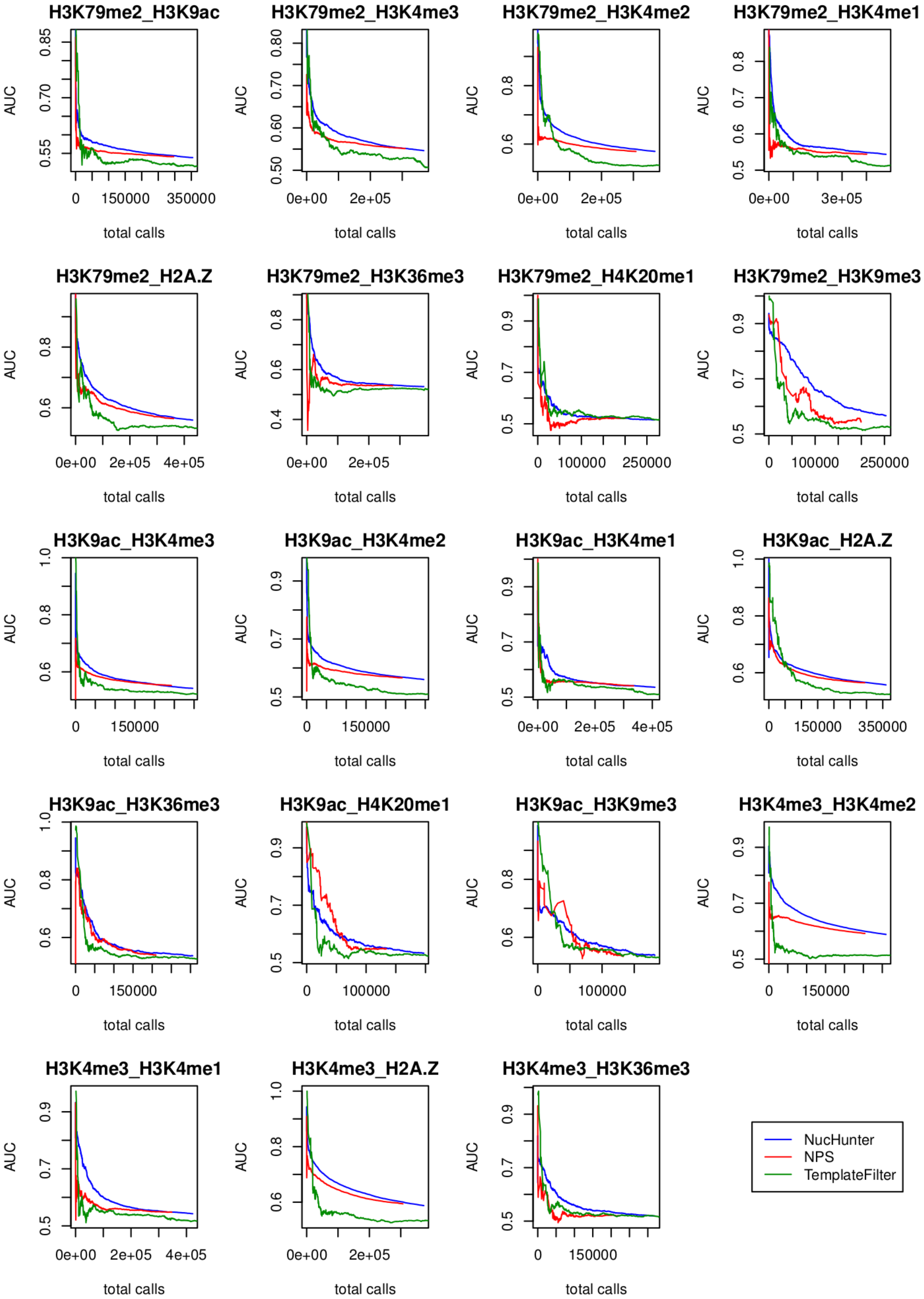

H3K27ac_H3K27me3
AUC
0.5
0.7
0.9
0
150000
total calls
H3K27ac_H3K79me2
AUC
0.6
0.8
0e+00
2e+05
total calls
H3K27ac_H3K9ac
AUC
0.6
0.7
0.8
0.9
0
150000
total calls
H3K27ac_H3K4me3
AUC
0.55
0.70
0.85
0
150000
total calls
H3K27ac_H3K4me2
AUC
0.6
0.8
0
150000
total calls
H3K27ac_H3K4me1
AUC
0.6
0.8
0e+00
2e+05
4e+05
total calls
H3K27ac_H2A.Z
AUC
0.6
0.8
0e+00
2e+05
total calls
H3K27ac_H3K36me3
AUC
0.6
0.8
1.0
0
150000
total calls
H3K27ac_H4K20me1
AUC
0.6
0.8
0
100000
total calls
H3K27ac_H3K9me3
AUC
0.6
0.8
0
100000
total calls
H3K27me3_H3K79me2
AUC
0.5
0.6
0.7
0.8
0.9
0e+00
2e+05
total calls
H3K27me3_H3K9ac
AUC
0.3
0.5
0.7
0.9
0
150000
total calls
H3K27me3_H3K4me3
AUC
0.4
0.6
0.8
1.0
0
150000
total calls
H3K27me3_H3K4me2
AUC
0.5
0.6
0.7
0.8
0
150000
total calls
H3K27me3_H3K4me1
AUC
0.5
0.7
0.9
0e+00
2e+05
4e+05
total calls
H3K27me3_H2A.Z
AUC
0.5
0.7
0.9
0e+00
2e+05
total calls
H3K27me3_H3K36me3
AUC
0.50
0.65
0.80
0
150000
total calls
H3K27me3_H4K20me1
AUC
0.4
0.6
0.8
0
100000
total calls
H3K27me3_H3K9me3
AUC
0.6
0.8
0
100000
total calls
NucHunter
NPS
TemplateFilter

Figure 8



H3K79me2_H3K9ac
AUC
0.55
0.70
0.85
0
150000
350000
total calls
H3K79me2_H3K4me3
AUC
0.50
0.60
0.70
0.80
0e+00
2e+05
total calls
H3K79me2_H3K4me2
AUC
0.6
0.8
0e+00
2e+05
total calls
H3K79me2_H3K4me1
AUC
0.5
0.6
0.7
0.8
0e+00
3e+05
total calls
H3K79me2_H2A.Z
AUC
0.6
0.8
0e+00
2e+05
4e+05
total calls
H3K79me2_H3K36me3
AUC
0.4
0.6
0.8
0e+00
2e+05
total calls
H3K79me2_H4K20me1
AUC
0.5
0.7
0.9
0
100000
250000
total calls
H3K79me2_H3K9me3
AUC
0.5
0.7
0.9
0
100000
250000
total calls
H3K9ac_H3K4me3
AUC
0.6
0.8
1.0
0
150000
total calls
H3K9ac_H3K4me2
AUC
0.5
0.7
0.9
0
150000
total calls
H3K9ac_H3K4me1
AUC
0.5
0.7
0.9
0e+00
2e+05
4e+05
total calls
H3K9ac_H2A.Z
AUC
0.6
0.8
1.0
0
150000
350000
total calls
H3K9ac_H3K36me3
AUC
0.6
0.8
1.0
0
150000
total calls
H3K9ac_H4K20me1
AUC
0.5
0.7
0.9
0
100000
total calls
H3K9ac_H3K9me3
AUC
0.6
0.8
0
100000
total calls
H3K4me3_H3K4me2
AUC
0.5
0.7
0.9
0
150000
total calls
H3K4me3_H3K4me1
AUC
0.5
0.7
0.9
0e+00
2e+05
4e+05
total calls
H3K4me3_H2A.Z
AUC
0.6
0.8
1.0
0e+00
2e+05
total calls
H3K4me3_H3K36me3
AUC
0.5
0.7
0.9
0
150000
total calls
NucHunter
NPS
TemplateFilter

Figure 8

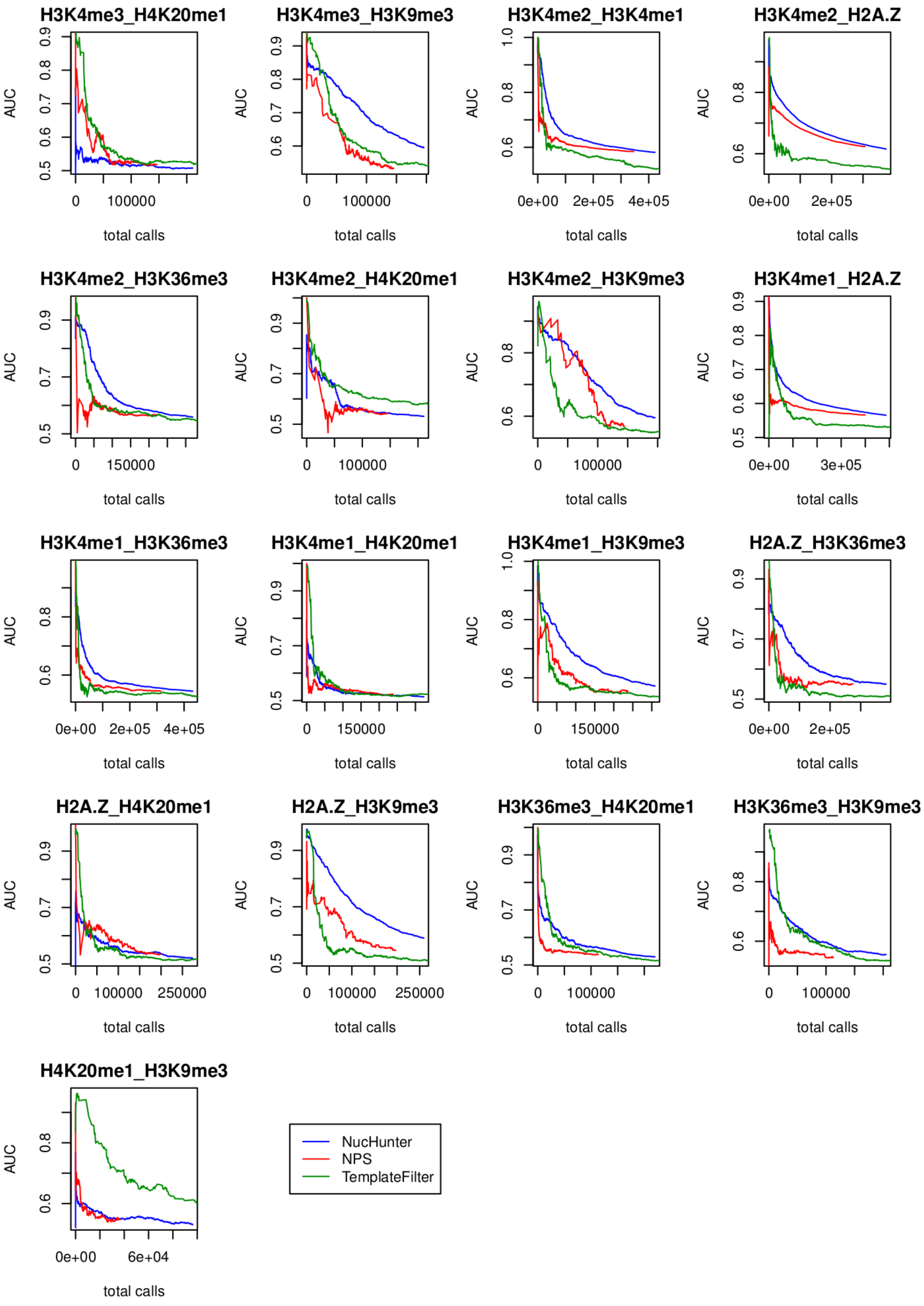


Figure 8: Performance of the three different algorithms on pairs of different ChIP-seq experiments. The end points of the curves show the total number of calls and the AUC using the default score thresholds.

| | File Info | | NucHunter | | NPS | | Template Filter | |
|---|---|---|---|---|---|---|---|---|
| chr | chr len | cov | rt | mem | rt | mem | rt | mem |
| chr1 | 249239814 | 3861846 | 26.42 | 1348768 | 96.43 | 2423296 | - | - |
| chr2 | 243074082 | 2847819 | 26.67 | 1444208 | 79.97 | 2360000 | - | - |
| chr3 | 197900591 | 2313629 | 21.27 | 1465232 | 62.84 | 1928800 | - | - |
| chr4 | 191043378 | 1633570 | 22.66 | 1355264 | 49.12 | 1858032 | - | - |
| chr5 | 180710750 | 1999720 | 20.60 | 1324016 | 55.80 | 1765568 | - | - |
| chr6 | 171010451 | 2394851 | 18.48 | 1422768 | 62.15 | 1677312 | - | - |
| chr7 | 159128122 | 1800691 | 19.61 | 1338192 | 49.26 | 1558736 | - | - |
| chr8 | 146299658 | 1479301 | 16.57 | 1373760 | 43.98 | 1437024 | - | - |
| chr9 | 141102566 | 1559164 | 18.20 | 1383456 | 44.42 | 1388688 | 673.71 | 26497584 |
| chr10 | 135513678 | 1611968 | 15.67 | 951728 | 45.92 | 1336608 | 680.39 | 25455408 |
| chr11 | 134945910 | 2198968 | 16.21 | 1368560 | 55.37 | 1335248 | 727.64 | 25351536 |
| chr12 | 133836731 | 2183800 | 14.64 | 1370624 | 52.41 | 1322928 | 700.99 | 25142928 |
| chr13 | 115107060 | 846871 | 11.96 | 1358784 | 26.79 | 1138464 | 445.55 | 21600272 |
| chr14 | 107288083 | 1339859 | 11.20 | 1319856 | 34.17 | 1067056 | 455.20 | 20135184 |
| chr15 | 102521084 | 1342934 | 11.16 | 1419760 | 35.84 | 1023664 | 448.22 | 19242160 |
| chr16 | 90277089 | 1472422 | 11.52 | 2350800 | 35.84 | 908720 | 397.43 | 16964608 |
| chr17 | 81194995 | 2193500 | 10.23 | 1443856 | 49.81 | 829760 | 461.92 | 15268208 |
| chr18 | 78016581 | 665533 | 9.37 | 1369936 | 21.02 | 787824 | 302.62 | 14653552 |
| chr19 | 59118844 | 2306566 | 8.69 | 2365888 | 45.08 | 620496 | 317.18 | 11123168 |
| chr20 | 62963996 | 943263 | 7.04 | 1407504 | 24.77 | 650016 | 265.72 | 11832672 |
| chr21 | 48101095 | 397273 | 6.33 | 1384976 | 12.75 | 504752 | 166.29 | 9035280 |
| chr22 | 51234688 | 771284 | 5.82 | 1382816 | 19.94 | 538096 | 192.81 | 9628944 |
| chrX | 154922080 | 774219 | 23.28 | 1415392 | 26.96 | 1512384 | - | - |
| chrY | 59030332 | 1389 | 10.26 | 759904 | 3.54 | 605728 | 181.62 | 11073680 |
| chrM | 16567 | 2536 | 0.64 | 291552 | 0.21 | 51200 | 0.05 | 9072 |

Table 1: Runtime and memory usage of the different algorithms on one-chromosome files derived from a H3K4me3 ChIP-seq experiment in human IMR90 cells. The column names have the following meaning: **chr** is the chromosome name, **chr len** is the chromosome length (spanned by reads), **cov** is the total read coverage, **rt** is the runtime (in minutes), **mem** is the maximum memory usage (in kilobytes). The symbol - means that the program crashed.

appropriately chosen. In order to test how stable the clustering procedure is for a given $k$, we ran the k-means algorithm 20 times with different initializations and we measured the degree of stability of the results.

To measure the consistency of a set of replicates, we defined a distance measure between two replicates and we considered the highest distance among all pairs.

Let $K_1, K_2 : \{1, 2, \dots n\} \rightarrow \{1, 2, \dots, k\}$ denote two classifications of $n$ objects into $k$ distinct classes, such as those provided by two different runs of the k-means algorithm on a dataset of $n$ vectors, and let the invertible function $\phi : \{1, \dots, k\} \rightarrow \{1, \dots, k\}$ denote a correspondence between the classes of the two classifiers. We define the misclassification error as:

$$d_\phi(K_1, K_2) = 1 - \frac{\sum_{i=1}^{k} |\{j \in \{1, 2, \dots n\} : K_1(j) = i \wedge K_2(j) = \phi(i)\}|}{n}$$

The correspondence function $\phi$ is chosen so as to minimize the misclassification error, so the distance between classifications $K_1$ and $K_2$ is: $d(K_1, K_2) = min_\phi\{d_\phi(K_1, K_2)\}$. Table 3 shows how the stability of the clustering algorithm varies with the number $k$ of clusters.

| **File Info** | | **NucHunter** | | **NPS** | |
|---|---|---|---|---|---|
| dataset | cov | rt | mem | rt | mem |
| H3K4me3 | 38942976 | 359.96 | 1204848 | 1028.87 | 2544096 |
| H3K9me3 | 43021873 | 239.43 | 2406320 | 7656.62 | 5235904 |
| H3K27ac | 43447140 | 388.21 | 1104496 | 2647.32 | 3408704 |
| H3K36me3 | 31779851 | 236.17 | 1920976 | 3697.70 | 3883136 |

Table 2: Runtime and memory usage of NucHunter and NPS on different ChIP-seq datasets in human IMR90 cells. The column names have the following meaning: **dataset** is the histone mark being analyzed, **cov** is the total read coverage, **rt** is the runtime (in minutes), **mem** is the maximum memory usage (in kilobytes).

| **k** | **mpd** |
|---|---|
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |
| 5 | 9.230E-05 |
| 6 | 0 |
| 7 | 5.680E-05 |
| 8 | 4.970E-05 |
| 9 | 7.881E-04 |
| 10 | 0 |
| 11 | 7.573E-05 |
| 12 | 2.674E-04 |
| 13 | 2.010E-01 |
| 14 | 1.417E-01 |
| 15 | 2.804E-03 |
| 16 | 1.713E-01 |

Table 3: Stability measure for the results of the k-means algorithm for different values of $k$. In the first column the number of clusters $k$ is specified, the second column specifies the maximum pairwise distance (mpd) between alternative clusterings on the same dataset. For small values of $k$ the results are particularly robust.

# 7 Genomic localization analyses

## 7.1 Average gene profile

For the average gene profile (see Figure 8 (a) in the main document) we selected annotated genes from RefSeq according to the following criteria:

1. we considered only genes with accession “NM” from the database,
2. we removed overlapping genes preferring those with more exons,
3. we filtered out genes whose transcript length was above the third or below the first quartile of the length distribution, which resulted in a set of 9408 genes.

Next, we computed nucleosome abundances at each position in the gene body. Since genes have different transcript lengths, we applied the following procedure to obtain an average rescaled profile:

1. we chose as a length $L$ for the average profile the median transcript length ($L = 21962$ bp)
2. let $x$ be the position of a nucleosome relative to the TSS of a transcript of length $K$. Its contribution needs to be located at a certain position $y$ relative to the TSS of the average profile.
    - if $x$ lies within a $[-5000, 2000]$ bp or a $K + [-2000, 5000]$ bp interval, then the nucleosome contributes one count and it is mapped respectively to positions $x$ or $L - K + x$ of the average profile.

- if $x$ lies between 2000 and $K - 2000$, then the nucleosome is mapped to position $y \simeq 2000 + (x - 2000)\frac{L-4000}{K-4000}$, where $y$ is rounded to an integer, and it contributes an adjusted count equals to $\frac{L-4000}{K-4000}$.

3. for displaying purposes, the obtained profile is smoothed with a running average

### 7.2 CAGE tags

Figure 8 (b) in the main document was obtained by considering a $[-2000, +2000]$ bp region around each CAGE tag, counting the nucleosome profile and computing the sum of all the profiles using the orientation given by the CAGE tag (so the profiles coming from CAGE tags mapped to the negative strands are flipped).

### 7.3 DNase tags

Figure 8 (c) in the main document was obtained by considering a $[-2000, +2000]$ bp region around each nucleosome, computing the DNase hypersensitivity profile, and computing the average across all regions.

## 8 Comparison with ChromHMM

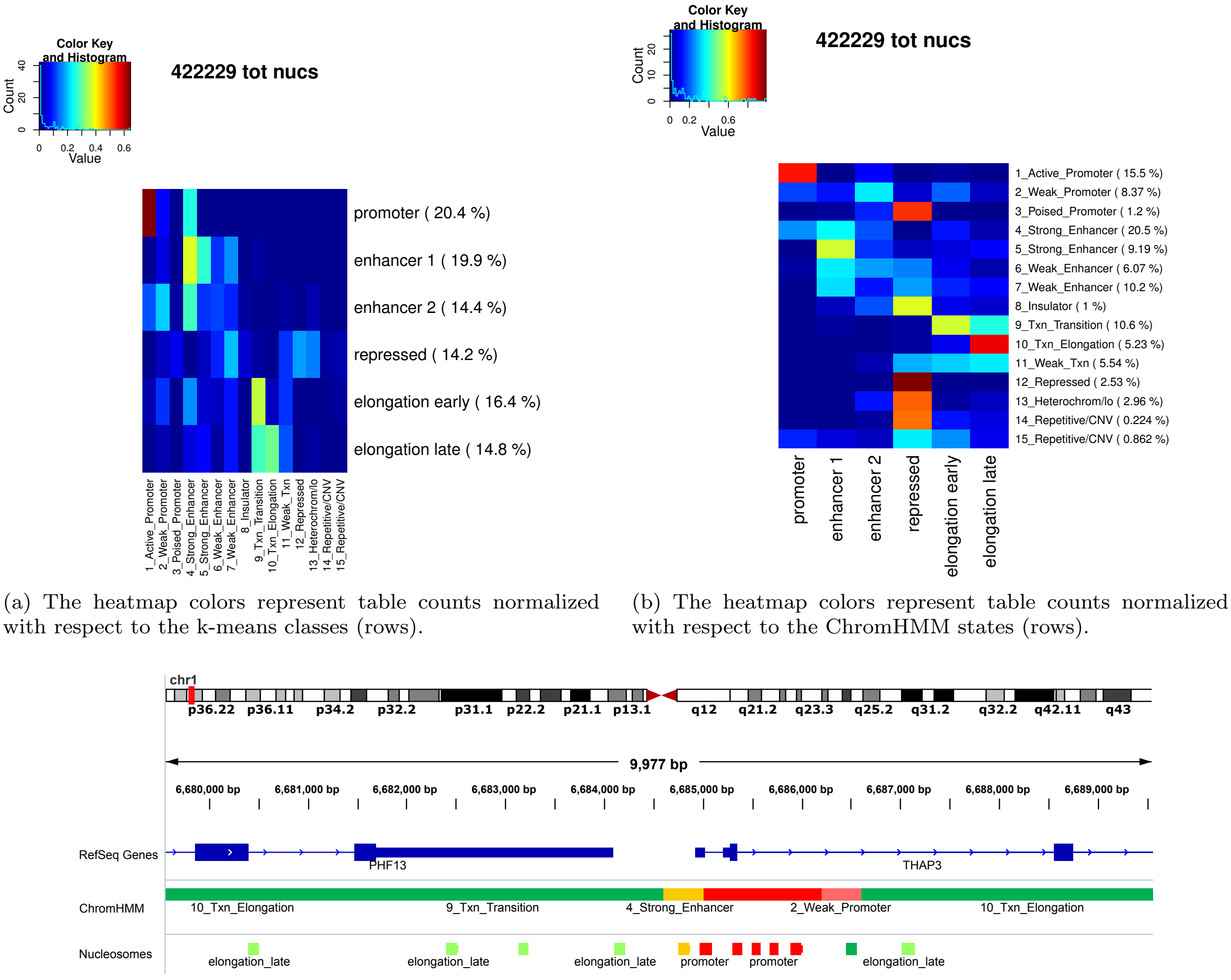
We compared the nucleosome classes obtained by clustering with the chromatin state classification provided by ChromHMM (Ernst and Kellis, 2010). We computed a table that counts for each pair $(c, s)$ the number of nucleosomes of class $c$ falling in a genomic region with chromatin state $s$, where $c$ ranges over the classes derived by k-means clustering and $s$ ranges over the states represented in ChromHMM. As it can be seen in Figure 9, the two annotations are in agreement, at least at a coarse scale, as promoter nucleosomes tend to be associated with promoter chromatin states, enhancer nucleosomes with enhancer states, elongation nucleosomes with transcription-associated states, and repressed nucleosomes with repressed states. The differences might be due, in addition to the different methods, also to the different data sources employed.

## References

Bernstein, B. E., Stamatoyannopoulos, J. A., Costello, J. F., *et al.* (2010). The NIH Roadmap Epigenomics Mapping Consortium. *Nat. Biotechnol.*, **28**(10), 1045–1048.

Ernst, J. and Kellis, M. (2010). Discovery and characterization of chromatin states for systematic annotation of the human genome. *Nat Biotech*, **28**(8), 817–825.

Landt, S. G., Marinov, G. K., Kundaje, A., *et al.* (2012). ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia. *Genome Res.*, **22**(9), 1813–1831.



(a) The heatmap colors represent table counts normalized with respect to the k-means classes (rows).

(b) The heatmap colors represent table counts normalized with respect to the ChromHMM states (rows).

(c) Screenshot showing the two alternative approaches to chromatin state analysis.

Figure 9: Correspondence between classes obtained from k-means clustering of nucleosome-associated histone modification read counts and chromatin states as inferred by the ChromHMM algorithm.